(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平10－232834

(43)公開日　平成10年(1998)９月２日

(51)Int.Cl.[6]	識別記号	FI	
Ｇ０６Ｆ　12/08		Ｇ０６Ｆ　12/08	Ｗ
			Ｇ
9/46	３４０	9/46	３４０Ｂ

審査請求　未請求　請求項の数10　ＯＬ　（全 11 頁）

(21)出願番号　特願平10－19476

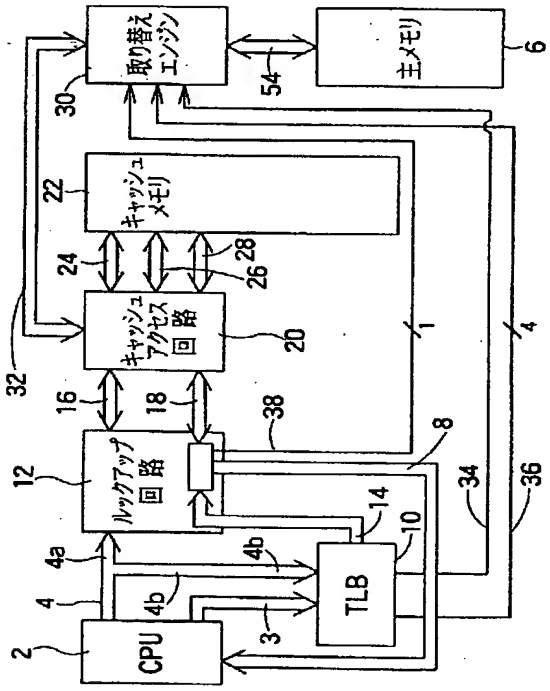
(22)出願日　平成10年(1998)１月30日

(31)優先権主張番号　９７０１９６０．８
(32)優先日　1997年１月30日
(33)優先権主張国　イギリス（ＧＢ）
(31)優先権主張番号　９７２５４３７．９
(32)優先日　1997年12月１日
(33)優先権主張国　イギリス（ＧＢ）

(71)出願人　595008364
エスジーエス－トムソン、マイクロエレクトロニクス、リミテッド
ＳＧＳ－ＴＨＯＭＳＯＮ　ＭＩＣＲＯＥＬＥＣＴＲＯＮＩＣＳ　ＬＴＤ.
イギリス国ブリストル、アーモンズベリー、アズテック、ウエスト、1000
(72)発明者　アンドルー、クレイグ、スタージス
イギリス国パース、ギニア、ランド、12エイ
(72)発明者　デイビッド、メイ
イギリス国ブリストル、クリフトン、イートン、クレスント、9
(74)代理人　弁理士　佐藤　一雄　（外３名）

(54)【発明の名称】　キャッシュメモリを作動する方法およびコンピュータシステム

(57)【要約】
【課題】　複数の並行処理を実行するプロセッサに対してより大きな予測可能性のあるキャシング動作をもたらすキャッシュシステムを提供すること。
【解決手段】　キャッシュメモリを、それぞれキャッシュメモリに項目を保持するためのアドレス指定可能な複数の記憶位置を有する複数のキャッシュ区分に分割し、キャッシュ区分のどれが当該処理の実行で使用するための項目を保持するために使用されるべきであるかを識別する区分表示子を各処理に割り当て、プロセッサが現在の処理の実行中に主メモリからの項目を要求し、かつこの項目がキャッシュメモリに保持されていない場合に、その項目を主メモリからフェッチし、かつ識別されたキャッシュ区分のアドレス指定可能な複数の記憶位置の中の１つにロードする。



【特許請求の範囲】
【請求項１】それぞれ命令のシーケンスを含む複数の処理を実行可能であるコンピュータのプロセッサと主メモリとの間に配置されたキャッシュメモリを作動する方法において、
前記キャッシュメモリを、それぞれキャッシュメモリに項目を保持するためのアドレス指定可能な複数の記憶位置を有する複数のキャッシュ区分に分割するステップと、
前記キャッシュ区分のどれが当該処理の実行で使用するための項目を保持するために使用されるべきであるかを識別する区分表示子を各処理に割り当てるステップと、
前記プロセッサが現在の処理の実行中に主メモリからの項目を要求し、かつこの項目が前記キャッシュメモリに保持されていない場合に、その項目を主メモリからフェッチし、かつ識別されたキャッシュ区分のアドレス指定可能な複数の記憶位置の中の１つにロードするステップとを含むことを特徴とする、キャッシュメモリを作動する方法。
【請求項２】現在実行されている現在処理のための前記区分表示子をメモリに保持するステップを有することを特徴とする請求項１に記載の方法。
【請求項３】新しい処理が前記プロセッサによって実行されるべきある場合、この新しい処理に割り当てられた新しい区分表示子が前記メモリにロードされることを特徴とする請求項２に記載の方法。
【請求項４】前記現在処理のための区分表示子を保持するメモリが前記処理についての状態情報をも保持する処理状態メモリであることを特徴とする請求項２または３に記載の方法。
【請求項５】前記区分表示子が前記処理のためのグループ識別子に含まれ、前記グループ識別子が前記処理のためのアドレス空間を識別することを特徴とする請求項１、２または３に記載の方法。
【請求項６】前記プロセッサが仮想ページ番号およびラインインページ番号を含むアドレスを発し、かつアドレス変換バッファが前記仮想ページ番号を、主メモリをアクセスする実ページ番号に変換するために備えられ、前記アドレス変換バッファが、前記グループ識別子をも受け取り、それから前記現在処理のための区分表示子を得ることを特徴とする請求項５に記載の方法。
【請求項７】各キャッシュ区分のアドレス指定可能な記憶位置数が変更可能であることを特徴とする請求項１ないし６のいずれかに記載の方法。
【請求項８】それぞれが一連の命令を含み、現在実行されている現在処理のための区分表示子を保持する処理状態メモリを含んでいる複数の処理を実行するためのプロセッサと、
主メモリと、
それぞれ処理の実行中のプロセッサによって使用するために前記主メモリからフェッチされる項目を保持するアドレス指定可能な複数の記憶位置を含む一連のキャッシュ区分を有するキャッシュメモリと、
前記主メモリから項目をフェッチし、この項目を前記アドレス指定可能な記憶位置のキャッシュメモリにロードするように構成され、現在処理と関連して処理状態メモリに保持された区分表示子に応じ前記項目をロードする前記アドレス指定可能な記憶位置の中の１つを選択するキャッシュ取り替え機構とを備えたことを特徴とするコンピュータシステム。
【請求項９】前記区分表示子が前記処理のためのアドレス空間を識別する各処理のためのグループ識別子に含まれていることを特徴とする請求項８に記載のコンピュータシステム。
【請求項１０】前記プロセッサが、仮想ページ番号およびラインインページ番号を含むアドレスを発し、かつ前記システムが前記仮想ページ番号を、主メモリをアクセスする実ページ番号に変換するアドレス変換バッファを備え、前記アドレス変換バッファが、前記グループ識別子をも受け取り、それから前記現処理のための区分表示子を得るように作動できることを特徴とする請求項９に記載のコンピュータシステム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、プロセッサとコンピュータの主メモリとの間で作動するキャッシュシステム、特に複数の並行処理を実行することができるプロセッサに関するものである。
【０００２】
【従来の技術】当該技術分野で周知のように、キャッシュメモリは、ある種のデータおよびコードへのアクセス待ち時間を減少させ、このデータおよびコードのために使用されるメモリバンド幅を減少させるようにコンピュータシステムにおいて使用されている。キャッシュメモリは、メモリアクセスを遅らせ、統合し、再配列することができる。
【０００３】キャッシュメモリは、コンピュータのプロセッサと主メモリとの間で作動する。プロセッサで実行するプロセスによって要求されるデータや命令は、この処理を実行している間、キャッシュに保持することができる。キャッシへのアクセスは、通常、主メモリへのアクセスよりも非常に高速である。プロセッサがキャッシュメモリに必要なデータ項目あるいは命令を置かなければ、プロセッサは、それを引き出すために主メモリを直接アクセスし、要求されたデータ項目あるいは命令はキャッシュにロードされる。キャッシュメモリを使用し、取り替えをするためのいろいろな公知のシステムが存在する。
【０００４】実時間システムのキャッシュを当てにするために、キャッシュの動作は予測できる必要がある。す

なわち、キャッシュにあることが予想される特定のデータ項目あるいは命令が実際そこにあることに関して適度の確実度がある必要がある。大部分の既存の取り替え機構は、要求されたデータ項目あるいは命令をキャッシュに入れようと通常、常に試みている。これを行うために、この機構はキャッシュから他のデータ項目あるいは命令を削除しなければならない。これは、後で使用するためにそこにあることが予想される、削除項目を生じ得る。これは、特に、多重タスクプロセッサ、あるいは割り込み処理あるいは他の予測不可能な処理を実効しなければならないプロセッサの場合である。

【０００５】

【発明が解決しようとする課題】本発明の目的は、複数の並行処理を実行するプロセッサに対してより大きな予測可能性のあるキャシング動作をもたらすキャッシュシステムを提供することにある。

【０００６】

【課題を解決するための手段】これに関しては、並行処理は、共有プロセッサによるが必ずしも同時に実行されない処理であるとみなされる。すなわち、第１の処理は実行し始め、何らかの理由で割り込むことができる。それから、プロセッサは、第２の処理を実行し始めるが、第１の処理が再び、あるいは何らかの他のプロンプトに応じていつでも実行できるようになったときに、第２の処理にいつでも割り込みできるようになっている。これは処理ハンドラによって管理される。第１の処理に関連したデータや命令が、第２の処理が実行している間にキャッシュから追い出されないことが重要である。反対に、第２の処理は、実行している間、キャッシュにアクセスすることができることが有用である。例えば、２つの処理、すなわち処理Ａおよび処理Ｂが１つのＣＰＵで同時に実行している図８に示された場合を考察する。処理Ａは最初にスケジュールされ、それがＣＰＵを保有している間、そのＣＰＵは、データキャッシュをその専用のデータで完全に満たすことができ、処理Ｂのためのデータキャッシュに入れられたいかなるデータをも追い出す。それから、制御は処理Ｂのためにスワッピングし、そのときデータキャッシュの状態を無効にし、処理Ａのデータの全てを取り除き、その専用のキャッシュに入れる。データキャッシュ状態のこのデータのピンポンは、並行処理との間で共有しており、しばしば性能上、有害である。

【０００７】請求項１に係る発明は、それぞれ命令のシーケンスを含む複数の処理を実行可能であるコンピュータのプロセッサと主メモリとの間に配置されたキャッシュメモリを作動する方法において、キャッシュメモリを、それぞれキャッシュメモリに項目を保持するためのアドレス指定可能な複数の記憶位置を有する複数のキャッシュ区分に分割するステップと、キャッシュ区分のどれが当該処理の実行で使用するための項目を保持するために使用されるべきであるかを識別する区分表示子を各処理に割り当てるステップと、プロセッサが現在の処理の実行中に主メモリからの項目を要求し、かつこの項目がキャッシュメモリに保持されていない場合に、その項目を主メモリからフェッチし、かつ識別されたキャッシュ区分のアドレス指定可能な複数の記憶位置の中の１つにロードするステップとを含むことを特徴とするものである。

【０００８】区分表示子を各処理に割り当てることによって、プロセッサで同時に実行している処理は、各々の他のデータや命令をキャッシュメモリから追い出すことが防止される。すなわち、例えばプロセッサで実行している第１の処理に割り当てられたキャッシュ区分はその後の第２の処理によって上書きすることができない。その代わりに、第２の処理は、それに割り当てられたそれの専用のキャッシュ区分を有する。一旦第１の処理が完全に完了すると、第１の処理に割り当てられたキャッシュ区分はそれから他の処理に割り当てることができるように、区分表示子を処理に割り当てることを変えることができることはもちろん好ましい。

【０００９】処理の要求に応じて、２つ以上のキャッシュ区分を処理に割り当てるか、あるいはキャッシュへの処理アクセスをともかく否定することができる。

【００１０】本発明の実施の形態においては、実行されている現在処理のための区分表示子は、処理についての状態情報をも保持する処理状態メモリに保持されている。これは、ここではスレッド状態ワードレジスタと呼ばれている。新しい処理がプロセッサによって実行されるべきである場合、新しいスレッド状態ワードは、この処理に割り当てられた新しい区分表示子とともにメモリにロードされる。

【００１１】区分表示子は、処理のためのグループ識別子の中に含むことができ、グループ識別子は処理のためのアドレス空間を識別する。仮想アドレス指定方式では、プロセッサは、仮想ページ番号およびページ番号のラインを含むアドレスを発し、アドレス変換バッファは、仮想ページ番号を主メモリをアドレス指定する実ページ番号に変換するために備えられている。このように、アドレス変換バッファは、グループ識別子も受け取り、処理に割り当てられた仮想アドレス空間に応じて現在処理のための区分表示子をそこから得ることができる。

【００１２】アドレス指定された項目のラインインページ番号は、項目が置かれるべきキャッシュ区分内のアドレス記憶位置を識別するために使用することができる。すなわち、各キャッシュ区分は直接マッピングされる。ラインインページ番号として項目アドレスのエンドビットであるが、適当なビットのセットに他ならない全てを使用することは必要ないことは明らかである。これらは、通常、アドレスの最下位の終わりに近い。

【００１３】１つあるいはそれ以上のキャッシュ区分を処理に割り当てることができる。

【００１４】このシステムは、キャッシュメモリからの項目を、前記項目の主メモリのアドレスに従って、かつ項目がキャッシュメモリに保持されるキャッシュ区分に関係なくアクセスするキャッシュアクセス回路を含み得る。すなわち、区分表示子は、取り替えで使用されるだけで、ルックアップでは使用されない。このように、キャッシュされる項目は、たとえこのキャッシング後にこの区分が異なるアドレス空間に関連した処理に今すぐ割り当てられるとしても、その区分から引き出すことができる。

【００１５】請求項８に係る発明は、それぞれが一連の命令を含み、現在実行されている現在処理のための区分表示子を保持する処理状態メモリを含んでいる複数の処理を実行するためのプロセッサと、主メモリと、それぞれ処理の実行中のプロセッサによって使用するために主メモリからフェッチされる項目を保持するアドレス指定可能な複数の記憶位置を含む一連のキャッシュ区分を有するキャッシュメモリと、主メモリから項目をフェッチし、この項目をアドレス指定可能な記憶位置のキャッシュメモリにロードするように構成され、現在処理と関連して処理状態メモリに保持された区分表示子に応じ項目をロードするアドレス指定可能な記憶位置の中の１つを選択するキャッシュ取り替え機構とを備えたことを特徴とするコンピュータシステムを要旨とするものである。

【００１６】各処理は、共有ページ番号内部に主メモリのアドレスに保持された２つ以上の命令シーケンスを含むことができる。キャッシュ区分は、各キャッシュ区分を主メモリの特定の処理のページ番号に関連付けることによって処理に割り当てることができる。これは、我々の以前の英国特許出願第９７０１９６０．８号明細書に記載されている。

【００１７】代替例として、区分表示子は、スレッド状態ワードレジスタに保持し、キャッシュ取り替え機構に直接供給することができる。

【００１８】各キャッシュ区分におけるアドレス指定可能な記憶位置の数は変更可能でありうる。さらに、ページ番号に対するキャッシュ番号の関連性は、変更可能でありうると同時に、これらのページ番号を使用するプロセスはプロセッサによって実行される。

【００１９】以下に記載する発明の実施の形態は、他の区分に入れられる他のデータのページからのキャッシュラインから読み出すかあるいはこのキャッシュラインに書き込むことによって予想できない追い出しからキャッシュの内容を保護するキャッシュシステムを示している。この実施の形態は、キャッシュの内容を予測できるシステムも提供する。

【００２０】

【発明の実施の形態】図１は、キャッシュシステムを組み込むコンピュータのブロック図である。コンピュータは、主メモリ６から項目をアクセスするためのアドレスバス４および項目をＣＰＵ２に戻すためのデータバス８に接続されているＣＰＵ２を備えている。データバス８はここではデータバスと呼ばれているが、この項目がＣＰＵによる実行のための実際のデータあるいは命令を構成しようが構成しまいが、これは主メモリ６からの項目の戻りのためであることが理解される。ここに記載されているシステムは命令キャッシュおよびデータキャッシュの両方で使用するのに適している。公知であるように、別個のデータキャッシュおよび命令キャッシュがあってもよいし、あるいはデータキャッシュおよび命令キャッシュは結合されてもよい。ここに記載されたコンピュータでは、アドレス指定方式はいわゆる仮想アドレス指定方式である。アドレスは、ページアドレスのライン４ａおよび仮想ページアドレス４ｂに分割される。仮想ページアドレス４ｂはアドレス変換バッファ（ＴＬＢ）１０に供給される。ページアドレスのライン４ａはルックアップ回路１２に供給される。アドレス変換バッファ１０は、仮想ページアドレス４ｂに基づいて変換された実ページアドレス１４をルックアップ回路１２に供給する。ルックアップ回路１２は、アドレスバスおよびデータバス１６，１８を介してキャッシュアクセス回路２０に接続されている。さらに、データバス１８は、主メモリ６からのデータ項目あるいは命令のためのものであってもよい。キャッシュアクセス回路２０は、アドレスバス２４、データバス２６およびキャッシュメモリ２２のための置換情報を転送する制御バス２８を介してキャッシュメモリ２２に接続されている。取り替えエンジン３０は、取り替えエンジンとキャッシュアクセス回路との間で置換情報、データ項目（あるいは命令）およびアドレスを転送する取り替えバス３２を介してキャッシュアクセス回路２０に接続されている。取り替えエンジン３０はそれ自体主メモリ６に接続されている。

【００２１】取り替えエンジン３０は、アドレス変換バッファ１０から、主メモリ６の項目の実ページアドレスおよびラインインページアドレスを含む全実アドレス３４を受け取る。取り替えエンジン３０は、４ビットバス上のアドレス変換バッファ１０から区分表示子をも受け取る。区分表示子の機能は以下に記載されている通りである。

【００２２】最後に、取り替えエンジン３０は、以下により明確に記載されているようにルックアップ回路１２で発生される、ライン３８上のミス信号を受け取る。

【００２３】ここに記載されているキャッシュメモリ２２は、直接マッピングキャッシュである。すなわち、このキャッシュメモリ２２は複数のアドレス指定可能な記憶位置を有し、各位置は、キャッシュの１行を構成する。各行は、主メモリからの項目およびこの項目の主メモリのアドレスを含んでいる。各行は、この行に記憶さ

れたデータ項目の主メモリのアドレスの最下位ビットを示す多数のビットによって構成される行アドレスによってアドレス指定可能である。例えば、８行を有するキャッシュメモリの場合、各行アドレスは、これらの行を独自に識別するための３ビットの長さである。例えば、キャッシュの第２行は行アドレス００１を有するので、ビット００１で終わるアドレスを有する主メモリからの任意のデータ項目を主メモリに保持することができる。明らかに、主メモリでは、多数のこのようなアドレスがあるので、可能性としてキャッシュメモリのこの行に保持される多数のデータ項目がある。もちろん、キャッシュメモリは、毎回この行に唯一のデータ項目を保持することができる。

【００２４】次に、図１に示されたコンピュータシステムの動作について説明するが、あたかも区分表示子が存在しないようである。ＣＰＵ２は、主メモリのアドレスを使用して主メモリ６からの項目を要求し、このアドレスをアドレスバス４上に伝送する。仮想ページ番号は、それを所定の仮想／実ページ変換アルゴリズムに従って実ページ番号１４に変換するアドレス変換バッファ１０に供給される。実ページ番号１４は、ＣＰＵ２によって伝送された最初のアドレスのページ番号のライン４ａとともにルックアップ回路１２に供給される。ラインインページアドレスは、キャッシュメモリ２２をアドレス指定するためにキャッシュアクセス回路２０によって使用される。ラインインページアドレスは、キャッシュメモリ２２の行アドレスに等しい、メモリの主アドレスの（必ずしもエンドビットを含まない）最下位数ビットのセットを含んでいる。データ項目（あるいは命令）およびデータ項目（あるいは命令）のアドレスである、ページアドレスの行によって識別される行アドレスのキャッシュメモリの内容は、ルックアップ回路１２に供給される。そこで、キャッシュメモリから引き出されたアドレスの実ページ番号は、アドレス変換バッファ１０から供給された実ページ番号と比較される。これらのアドレスが一致するならば、ルックアップ回路はヒットを示し、このヒットによってキャッシュメモリのこの行に保持されたデータ項目はデータバス８に沿ってＣＰＵに戻される。しかしながら、キャッシュメモリ２２のアドレス指定された行に保持されたアドレスの実ページ番号がアドレス変換バッファ１０から供給された実ページ番号に一致しないならば、ミス信号は取り替えエンジン３０のライン３８上に発生される。バス３４上のアドレス変換バッファ１０から供給される実アドレスを使用して、主メモリ６から正しい項目を引き出すことは取り替えエンジン３０のタスクである。主メモリ６から１度フェッチされたデータ項目は、取り替えバス３２を介してキャッシュアクセス回路２０に供給され、主メモリのアドレスとともにキャッシュメモリ２２にロードされる。データ項目そのものはデータバス８に沿ってＣＰＵにも戻されるので、ＣＰＵは実行し続けることができる。上記に概略されるような直接マッピングキャッシュメモリでは、主メモリ６から呼び出されるデータ項目およびそのアドレスは、データ項目がチェックするために最初にアクセスされた記憶位置にロードされることは明らかである。すなわち、データ項目は、主メモリのラインインページアドレスの最下位数ビットのセットに一致する行アドレスを有するデータ項目を受け入れることができる唯一の記憶位置に上書きされる。もちろん、キャッシュメモリに最初に記憶されたデータ項目のページ番号およびいまキャッシュメモリにロードされるべきデータ項目は異なっている。この「１対１のマッピング」はキャッシュの有用性を制限する。

【００２５】より大きな柔軟性をキャッシュシステムに提供するために、ｎウェイ・セット・アソシアティブ・キャッシュが開発された。４ウェイ・セット・アソシアティブ・キャッシュは図２に示されている。キャッシュメモリは４つのバンクＢ１，Ｂ２，Ｂ３，Ｂ４に分割される。このバンクは、図２の１行に対して概略的に示されているように、一般に共有行アドレスによって行方向にアクセスされる。しかしながら、この行は４つのキャッシュエントリ、すなわち各バンクに対して１つのキャッシュエントリを含んでいる。バンクＢ１に対するキャッシュエントリは、バス２６ａ上に出力され、バンクＢ２に対するキャッシュエントリは、バス２６ｂ上に出力され、バンクＢ３およびＢ４に対しても同様である。したがって、これは、１行アドレス（あるいはページアドレスの行）に対して４つのキャッシュエントリを可能にする。行がアドレス指定される度に、４つのキャッシュエントリが出力され、そのアドレスの実ページ番号は、アドレス変換バッファバッファ１０から供給された実ページ番号と比較され、どのエントリが正しいエントリであるかを決定する。キャッシュに試みられたアクセスの際にキャッシュミスがあるならば、取り替えエンジン３０は、主メモリから要求された項目を引き出し、例えば、特定の項目がどれくらいの長さキャッシュに保持されているか、あるいはシステムの他のプログラムパラメータに基づいている取り替えアルゴリズムに従って、バンクの中の１つの正しい行に要求された項目をロードする。このような置換アルゴリズムは公知であり、ここではこれ以上の説明は省略する。

【００２６】それにもかかわらず、ｎウェイ・セット・アソシアティブ・キャッシュ（ここでは、ｎはバンク数であり、図２の４に等しい）は、単一の直接マッピングシステムの改良であるかぎり、なお汎用性がなく、より重要なことはキャッシュの動作を適切に予測できないことである。

【００２７】ここに記載されたシステムは、より汎用性のあるキャッシュ取り替えシステムによってキャッシュメモリのコンピュータ使用の最適化を可能にするキャッ

シュ分割機構を備えている。

【００２８】図3は、図1のコンピュータを使用するＣＰＵ２の概略ブロック図である。ＣＰＵ２は、メモリバス４を介してメモリをアドレス指定し、データバス８を介してデータおよび命令を引き出すことができるフェッチ回路１７に接続されている実行回路１５を備えている。汎用レジスタ７のセットは、プロセスを実行する際に使用するためのデータおよび命令を保持する実行回路１５に接続されている。さらに、参照番号９、１１および１３によって示される特別レジスタのセットが備えられている。任意の数の専用レジスタがあってもよく、例として、レジスタ１１は、現在実行されているコードのラインを識別する命令ポインタを保持する。さらに、特殊レジスタ９は、ＣＰＵ２によって実行されているプロセスの状態を規定するスレッド状態ワードを保持する。実行回路１５は、毎回１プロセスあるいは命令のシーケンスを実行することができる。しかしながら、実行回路１５は、同様にこのプロセスに割り込み、第１のプロセスが実行し終える前に別のプロセスを実行し始める。実行回路１５によって現在実行されるプロセスに割り込むことができる多くの理由がある。１つは、より高い優先順位の割り込み処理が直ちに実行されることである。もう一つは、実行されている処理が現在長い待ち時間でデータを待っているので、第１の処理がこのデータを待っている間、その後の処理を実行し始めることは実行回路にとってより効率的であることである。データが受け取られた場合、第１のプロセスを実行するために再スケジュールすることができる。並行処理の実行は、それ自体は公知であり、処理ハンドラ１９によって管理される。

【００２９】各処理は、いわゆる制御の「スレッド」の下で実行される。スレッドは下記の状態を有する。すなわち、この処理において、スレッドがどこに進むかを示している命令ポインタ、この処理が次にどこに分岐するかを示すジャンプポインタ、直ちにアクセスできる値を含んでいる汎用レジスタ７のセット、仮想アドレス／物理アドレスのマッピング、仮想アドレスによりアクセス可能なメモリの内容、および、スレッドおよび任意の他の値によってこのような浮動小数点丸めモード、スレッドがカーネル特権を有しているかどうか等をアクセスできる制御レジスタ。

【００３０】上記の状態のいくつかは、ここではスレッド状態ワードと呼ばれ、スレッド状態ワードレジスタ９に保持されている小さい値のセットで指定される。スレッド状態ワードは、特に下記についての情報を保持する。スレッドがカーネルモードにあるか否か、スレッドはどの仮想アドレス空間にアクセスできるか、浮動小数点フラグ、トラップイネーブルおよびトラップモード、デバッグ情報、およびトラップ最適化情報。

【００３１】スレッド状態ワードのフォーマットは表Ｉに規定されている。

【表１】

表 I

名　前	ビット*	サイズ	説　明
TSW.FPFLAG	0～7	8	浮動小数点例外フラグ
TSW.FPTRAP	8～15	8	浮動小数点例外トラップ
TSW.FPMODE	16～19	4	浮動小数点モード
	20～31		予備
TSW.USER	32	1	カーネルモード（0）ユーザモード（1）
TSW.SINGLE	33	1	単一ステップモード
TSW.TLB	34	1	第1のレベルＴＬＢミスハンドラ表示子
TSW.WATCH	35	1	イネーブルされた監視ポイント
TSW.ENABLE	36	1	トラップイネーブル
	37～47	11	予備
TSW.GROUP	48～55	8	グループ番号
	56～63		予備

【００３２】表Ｉから分かるように、スレッド状態は、８ビットのグループ番号を含んでいる。これは、キャッシュ区分を割り当てる区分表示子を発生するために下記に記載されるように示される。

【００３３】ここに記載されたシステムのアドレス変換バッファ１０では、各ＴＬＢエントリは、仮想ページ番号、実ページ番号および情報シーケンスに関連付けた。情報シーケンスは公知であり、ここにこれ以上説明されていない方法でメモリのアドレスについてのいろいろな情報を含んでいる。しかしながら、現在説明されているシステムによれば、情報シーケンスは、グループ番号および仮想ページ番号によって決まる区分表示子を発生する区分コードをさらに含んでいる。これは、図4に概略図で示されている。ここでは、ＶＰは仮想ページ番号を

示し、ＲＰは実ページ番号を示し、ＧＮがグループ番号を示し、ＩＮＦＯが情報シーケンスを示している。記載されている実施の形態では、ＰＩは４ビットの長さである。

【００３４】したがって、情報シーケンスＩＮＦＯのビット０～３は区分表示子を構成する。区分表示子は、データ項目がキャッシュメモリ２２に最初にロードされるときにデータ項目を入れることができる区分に関する情報を示している。図２に示されたキャッシュ構造に関して、各区分はキャッシュの１バンクを構成することができる。区分表示子において、各ビットはバンクの中の１つを参照する。区分表示子のビットｊの１の値は、このページのデータを区分ｊに入れることができないことを意味する。ビットｊの０の値は、このページのデータを区分ｊに入れることができることを意味する。データは、区分表示子の２ビット以上に０を有することによって２つ以上の区分に入れることができる。全て０である区分表示子によって、データはキャッシュの任意の区分に入れることができる。全て１である区分表示子によって、任意のデータ項目は、キャッシュメモリにロードすることができない。これは、例えば、キャッシュの内容を“凍結”するために、例えば診断目的のために使用することができる。

【００３５】図４に示された例では、区分表示子は、主メモリにこの実ページ番号を有するデータ項目の置換がバンクＢ１あるいはＢ３を使用することができず、バンクＢ２あるいはＢ４が使用可能であることを示している。

【００３６】２つ以上のバンクをページに割り当てることは事実上可能である。この場合、ラインインページアドレスがキャッシュに対して行アドレスよりも多くのビットを有するならば、区分はｋウェイ・セット・アソシアティブ・キャッシュとして動作する。ここで、ｋ個の区分は１ページに割り当てられる。したがって、記載された例では、図４の処理はバンクＢ２およびＢ４を使用することができる。しかしながら、この処理はバンクＢ１およびＢ３を使用することができない。

【００３７】区分情報は、キャッシュルックアップで使用されないで、キャッシュ置換あるいは取り替えにだけ使用される。このように、キャッシュアクセスは、キャッシュメモリのどこにも保持されるデータ項目を探索できるのに対して、置換は、このページアドレスのために許可された区分にデータを置換するだけである。

【００３８】図５は、取り替えエンジン３０の内容をより詳細に示している。取り替えバス３２は、３つの別個のバス、データバス３２ａ、アドレスバス３２ｂおよび置換情報を伝達するバス３２ｃとして図４に示されている。データバスおよびアドレスバス３２ａおよび３２ｃは、メモリバス５４を介して主メモリをアクセスするメモリアクセス回路５０に供給される。置換情報は、実アドレス３４、バス３６上の区分表示子ＰＩおよびミス信号３８も受け取る決定回路５２に供給される。決定回路５２は、主メモリにアクセスされたデータが置かれるべきキャッシュの適切な区分を決定する。

【００３９】ここに記載されたキャッシュ分割機構は特に多重タスクＣＰＵに有用である。多重タスクプロセッサは、２つ以上の処理を「同時に」実行することができる。実際には、プロセッサは処理の一部を実行し、この処理が何かの理由で、多分続けるためのデータあるいはスティミュラスの要求で停止されると、プロセッサは直ちに他の処理を実行し始める。したがって、プロセッサは、個別処理を続けるためのデータあるいはスティミュラスを待って中止することができるときさえ常に作動している。図６は、このような状態を概略図で示している。図６の左側には、プロセッサが異なるＰ１，Ｐ２，Ｐ３，Ｐ４を実行することを引き受けることができるシーケンスが示されている。これらの処理が、そのデータがメモリに保持されることを予想できる場合の図が、図５の右側に示されている。したがって、処理Ｐ１のためのデータはページ０に保持されている。処理Ｐ２のためのデータはページ１およびページ２に保持されている。処理Ｐ３およびＰ４のためのデータはページ３を共有する。この例では、プロセッサは、第１の処理のシーケンスＰ１、第１の処理のシーケンスＰ２、第２の処理のシーケンスＰ１、第２の処理のシーケンスＰ２を実行し、それから第１の処理のシーケンスＰ３を実行する。第２の処理のシーケンスＰ１が実行された場合、処理Ｐ１はプロセッサによって完全に実行された。従来のキャッシュシステムでは、一旦プロセッサが第１の処理のシーケンスＰ２を実行し始め、このようにページ１からのアクセスを要求すると、これらのラインのデータ項目および命令は０ページからの予め記憶された記憶項目および命令をキャッシュで置換することは容易に明らかである。しかしながら、、第２の処理のシーケンスＰ１が実行されると、これらはまもなく再び要求することができる。

【００４０】ここに記載されているキャッシュ分割機構は、これから生じる得るタイミング遅れおよび不正確さを避ける。図７は、プロセッサが処理Ｐ１を実行している間のキャッシュの分割およびプロセッサが実行Ｐ３等にスイッチする場合の分割の変化を示している。図６は、各々の場合に対するＴＬＢキャッシュ区分表示子も示している。したがって、図５は、プロセッサが処理Ｐ１およびＰ２を実行している間の分割されたキャッシュを左側に示している。処理Ｐ１はキャッシュのバンクＢ１およびＢ２を使用することができるが、バンクＢ３およびＢ４を使用することはできない。反対に、処理Ｐ２はバンクＢ３およびＢ４を使用することができるが、バンクＢ１およびＢ２を使用することはできない。これはＴＬＢエントリで分かる。すなわち、処理Ｐ１は、バンクＢ１およびＢ２をアクセスすることができるが、バン

クＢ３およびＢ４にアクセスできないキャッシュ区分表示子を有する。処理Ｐ２は、バンクＢ３およびＢ４にアクセスすることができるが、バンクＢ１おＢ２にアクセスできないキャッシュ区分表示子を有する。処理Ｐ３は、キャッシュにアクセスすることを防止するキャッシュ区分表示子を有する。したがって、処理Ｐ３からのデータ項目をキャッシュにロードしようとするプロセッサによるいかなる試みも禁止される。しかしながら、前述の処理シーケンスに関して、理解できるように、プロセッサは、処理Ｐ１を実行し終えるまで、処理Ｐ３のいかなる部分も実行するつもりはないために、これは不利なことではない。プロセッサが何らかの理由でＰ３を実行しなければならないならば、唯一の不利な面は、プロセッサがダイレクトメモリからそのアクセスを行わなければならなく、キャッシュを使用できないことである。

【００４１】処理Ｐ１が実行し終えた場合、プロセッサは、キャッシュ区分表示子を変えることができるようにカーネルモードを要求することができる。これが行われる方法は、分割機構が実行される方法で決まる。前述の例の場合、区分コードは任意の他のＴＬＢエントリと同様にＴＬＢにセットすることができる。このように、区分コードはＣＰＵ２で実行するカーネルモードソフトウェアによって通常セットされる。しかしながら、ユーザは、キャッシュ区分が変えられるべきであることを要求することによって区分を変えることができる。この場合、ＣＰＵ２は、要求を実行するためにカーネルモードに変え、それに応じてＴＬＢエントリを変え、それからユーザモードに戻し、ユーザは継続することができる。したがって、ユーザは、キャッシュの分割動作を変えることができるので、これまで可能であったよりも非常に大きな柔軟性をもたらす。この変化は図６の右側に示されている。したがって、次にキャッシュ区分表示子は、処理Ｐ１がキャッシュをともかく使用することを防止するが、キャッシュ区分表示子がキャッシュのこれらのバンクにアクセスできるように処理Ｐ３およびＰ４のためのキャッシュ区分表示子を変えることによってバンクＢ１およびＢ２を処理Ｐ３およびＰ４に割り当てる。このように、プロセッサが処理Ｐ３を実行することを予想している場合、プロセッサは今やキャッシュ機構を有する。

【００４２】したがって、並行処理がデータキャッシュから他の各データを追い出すことを防止するシステムが前述された。すなわち、この処理はデータキャッシュ区分をばらばらにするようにマッピングされる。これは、それの独自の専用データキャッシュに各処理を有効的にもたらす。これは両方の処理に使用可能なデータキャッシュ空間量を減少させるが、それによってその動作は正確に予測するのが非常に容易になる。ここに記載されているシステムの結果は図９に示されている。

【００４３】ここに記載されているシステムが特に有用である他の領域は性能に重要なルーチンの実行にある。その性能がシステムの全性能に絶対的に重要である若干のルーチンがある。これの適切な例は、呼び出される場合、保証された（通常短い）時間の長さに効果を生じなければならない割り込みサービスルーチンであってもよい。これらの場合、キャッシュ区分は、これらの重要なルーチンに必要なデータおよびコードのためのデータキャッシュおよび命令キャッシュの両方に取っておかれる。したがって、命令キャッシュおよびデータキャッシュの残りは残りの処理の中で分配することができる。図１０は、可能な配置を示している。図１０では、データキャッシュの１つの２キロバイトおよび命令キャッシュの１つの４キロバイトを動作に重要な割り込みサービスルーチンのために取っておく例が示されている。

【００４４】本発明が上記に詳述された実施の形態に限定されないことは明らかである。いくつかの特定の可能な変更は後述されるが、これは本発明の範囲内で可能である包括的な変更のリストではない。

【００４５】上記の実施の形態では、アドレスバス４上のＣＰＵによって発されるアドレスは、仮想ページ番号４ｂおよびラインインページ４ａに分割される。しかしながら、本発明は、全仮想アドレスがＣＰＵからキャッシュのためのルックアップ回路に送られている場合に使用することもできる。反対に、本発明は、ＣＰＵが実アドレスを直接にルックアップ回路に発する場合にも適用可能である。重要なことは、キャッシュ区分表示子が実行されている特定の処理と関連して提供されていることである。

【００４６】上記の実施の形態では、単一のキャッシュアクセス回路２０は、ルックアップおよび取り替えの両方に関してキャッシュをアクセスするように示されている。しかしながら、キャッシュに取り替えのための付加アクセスポートを装備することも可能なので、ルックアップおよび取り替えはキャッシュメモリ２２のための異なるアクセスポートによって行われる。

【００４７】上述の実施の形態では、取り替えエンジン３０およびキャッシュアクセス回路２０は個別ブロックで示されている。しかしながら、その機能を組み合わせてルックアップおよび取り替えの両方を実行する単一のキャッシュアクセス回路にすることも可能である。

【００４８】下記は、特定の処理と関連して区分表示子を発生するのに可能な代替例を示すものである。

【００４９】１つの代替例では、区分表示子は、スレッド状態ワードＴＳＷに直接入れられる。記載されたスレッド状態ワードの場合、これは、ＴＳＷの現在予備の５６～５９を新しいフィールドＴＳＷ．ＰＩに割り当てることによって行うことができる。それから、ＴＳＷ．ＰＩの値は、ＣＰＵから直接取り替えエンジン３０に送られる。これは、ＴＬＢ１０からよりもむしろＣＰＵ２から直接区分表示子を取り替えエンジン３０に接続するた

めに図１に示されたアーキテクチャの修正を必要とする。これを実施するために、新しいスレッド状態ワードＴＳＷは実行される次のスレッドのためにロードされる場合、区分表示子ＰＩが変えられる。これは、スレッド状態ワードのパラメータをセットする特定の設定命令で行うことができる。

【００５０】他の実施の形態では、仮想アドレスを使用しないでグループ番号を区分表示子ＰＩにマッピングするアドレス変換バッファ１０にテーブルを備えることができる。これは、区分表示子をもたらすグループ番号によって索引付けられたテーブルあるいは一致グループのための区分表示子をもたらすグループ番号／区分表示子対を有するテーブルを備えることによって行うことができる。この場合、図１のアーキテクチャは変わらないが、異なるテーブルはアドレス変換バッファ１０で必要である。これを実施する場合、区分表示子は、２つのオペランド、すなわち変えられる制御レジスタ番号およびその新しい値を有する“ｐｕｔ”命令を使用して変えることができる。全ての制御レジスタは、この命令を使用してこのレジスタにアクセスするために使用することができる番号を割り当てられるので、グループ番号／区分表示子テーブルの各エントリは独自の制御レジスタ番号を有する。

【００５１】

【発明の効果】本発明によれば、複数の並行処理を実行するプロセッサに対してより大きな予測可能性のあるキャシング動作をもたらすキャッシュシステムを提供することができる。

【図面の簡単な説明】

本発明をより完全に理解し、同様に本発明を実行できる方法を示すために、いま例として添付図面が参照される。

【図１】キャッシュシステムを組み込むコンピュータのブロック図である。

【図２】４ウェイ・セット・アソシアティブ・キャッシュを示す略図である。

【図３】図１のＣＰＵのブロック図である。

【図４】アドレス変換バッファのエントリの一例である。

【図５】取り替えエンジンのブロック図である。

【図６】多重タスクプロセッサの動作を示す図である。

【図７】図６のシステムのためのキャシング動作の変更を示す図である。

【図８】未分割キャッシュを示す図である。

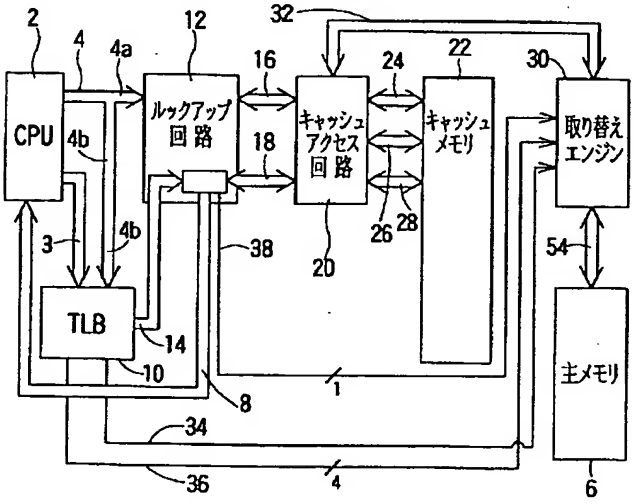
【図９】本発明の有用なアプリケーションを示す図である。

【図１０】本発明の有用なアプリケーションを示す図である。

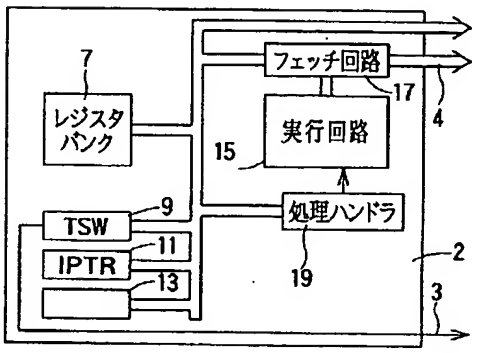
【符号の説明】

２　ＣＰＵ
６　主メモリ
７　レジスタバンク
１０　アドレス変換バッファ（ＴＬＢ）
１２　ルックアップ回路
１５　実行回路
１７　フェッチ回路
１９　処理ハンドラ
２０　キャッシュアクセス回路
２２　キャッシュメモリ
３０　取り替えエンジン
５０　メモリアクセス回路
５２　決定回路

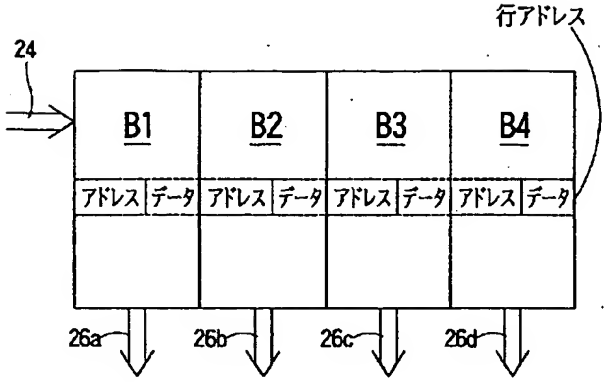
【図１】



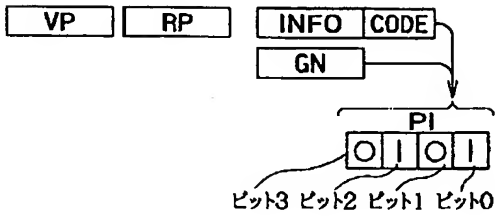
【図３】



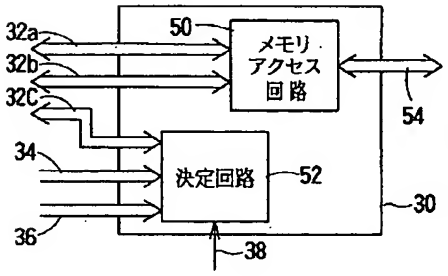
【図2】



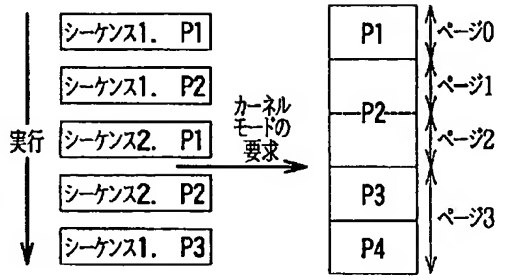
【図4】



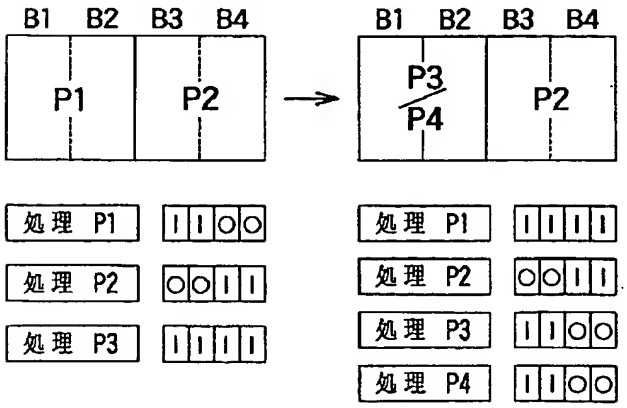
【図5】



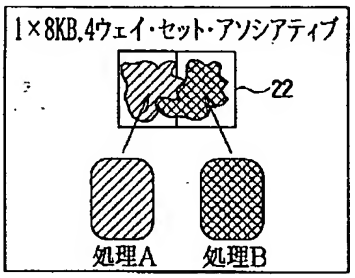
【図6】



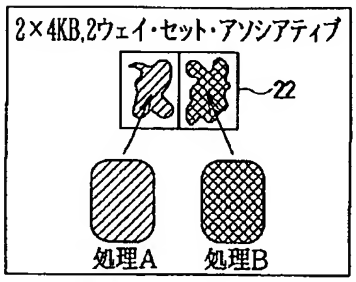
【図7】



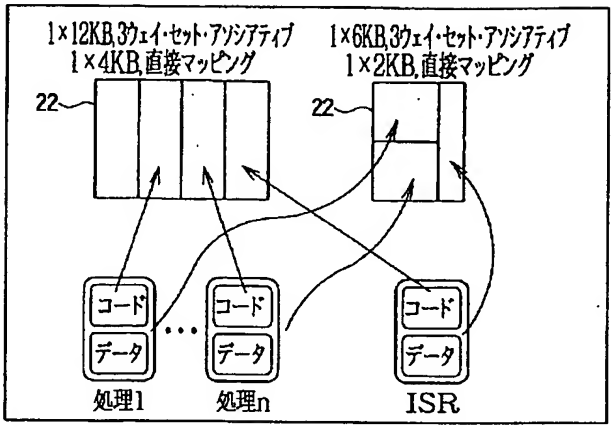
【図8】



【図9】



【図10】


1×12KB,3ウェイ・セット・アソシアティブ
1×4KB,直接マッピング
1×6KB,3ウェイ・セット・アソシアティブ
1×2KB,直接マッピング
22
22
コード
データ
コード
データ
コード
データ
処理1
処理n
ISR